WHAT IS CLAIMED IS:

1. A method for mapping a texture onto a surface of a computer generated object comprising the steps of:
approximating a true pixel color by performing a number of texturing operations, said texturing operations being determined by a geometric shape of a projection of a pixel on the texture; and
averaging results of said texturing operations.

2. A method as set forth in claim 1, wherein each of said texturing operations comprises:

accessing a mipmap at least one time; and

responding to multiple accesses being performed by interpolating results of the accesses.

3. A method as set forth in claim 2, wherein said number of texturing operations is a power of two.

4. A method as set forth in claim 3, wherein said number of texturing operations is less than or equal to a predetermined limit.

5. A method as set forth in claim 2, wherein the texture represents a reflected environment.

6. A method as set forth in claim 2, further comprising:

modifying a specularly reflected light intensity on the surface by combining said specularly reflected light intensity with a specular reflectance coefficient, said specular reflectance coefficient being retrieved from a specular reflectance coefficient map associated with the surface.

7. A method for modifying a specularly reflected light intensity on a surface of a computer generated object, comprising:

combining the specularly reflected light intensity with a specular reflectance coefficient, said specular reflectance coefficient being retrieved from a specular reflectance coefficient map associated with the surface.

8. A method for adding detail to a texture map comprising at least one texture element, the method comprising the steps of:

generating a detail map; and

assigning a pointer into said detail map to at least one of the texture elements of the texture map to generate a pointer map, said pointer comprising two offsets.

9. A method as set forth in claim 8, wherein said detail map is organized as a mip-map.

10. A method as set forth in claim 9, further comprising the steps of:

determining a texture address and a level of detail;

responding to said level of detail indicating that detail is needed by, retrieving offsets from said pointer map;

using said offsets as detail map addresses;

accessing said detail map;

responding to multiple accesses of the detail map by, interpolating results of the detail map accessing; and

mapping the texture map and the detail map to a surface of a computer generated object.

11. A method as set forth in claim 10, wherein a final pixel color is a combination of the results of the detail map access operation and a texture map access operation, said texture map access operation comprising at least one access to the texture map.

12. A method as set forth in claim 11, wherein at least one of said texture map access operation, and said detail map access operation is carried out in real time using dedicated arithmetic units.

13. A device for generating a texture map, environment map, reflectance map and detail map, comprising:

a memory unit for storing at least one of a texture map, an environment map, a reflectance map, and a detail map; and

a dedicated arithmetic unit, responsive to said memory unit, for generating at least one of said texture map, environment map, reflectance map, and detail map.

14. The device of claim 13, further comprising:

a filter unit for generating prefiltered images of less detail; and means for accessing pixels of a previous half-frame to perform said filtering.

15. A device for mapping interlaced real time video images onto a surface of a computer generated object, each video image including two interlaced half-frames of pixels, comprising:

a filter unit for generating prefiltered images of less detail; and

means for accessing pixels of a previous interlaced half-frame to perform said filtering.

16. A method for mapping a texture onto a surface of a computer generated object represented by a plurality of pixels, comprising the steps of :

dividing a texture map into blocks, the texture map comprising a plurality of texels, each texel having an associated value;

determining two block values for each block, which block values are representative of the values of texels in the block;

compressing the texture map by assigning to each texel one of the block values associated with the block of which it is part; and

mapping said compressed texture map onto the surface of the computer generated object.

17. A method as set forth in claim 16, wherein the block values associated with the texture map are quantized to a smaller number of bits.

18. A method as set forth in claim 16, wherein the step of determining two block values for each block comprises:

calculating a tensor of inertia from texel values;

determining an eigenvector having a smallest eigenvalue from said tensor;

multiplying said smallest eigenvalue eigenvector with said texel values; and

splitting the texel values in two groups by comparing a result of said multiplication with a threshold value.

19. A method as set forth in claim 16, wherein the texture map corresponds to a filtered texture map of lesser detail than a texture map of full detail.

20. A method as set forth in claim 16, wherein the step of mapping said compressed texture map onto the surface of the computer generated object comprises:

for each pixel which represents the computer generated object,

accessing said compressed texture map at least one time; and

responding to said compressed texture map being accessed more than one time by, interpolating results of the accesses.

21. A method as set forth in claim 20, wherein the step of mapping said compressed texture map onto the surface of the computer generated object further comprises:

approximating true pixel color by performing a number of texturing operations according to a geometric shape of a projection of a pixel on the texture and averaging results of said texturing operations.

22. A method as set forth in claim 21, wherein the texture is an environment map.

23. A method as set forth in claim 22, wherein at least one of said texture mapping, environment mapping, reflectance mapping and detail mapping is carried out in real time using dedicated arithmetic units.

24. A device for at least one of texture mapping, environment mapping, reflectance mapping and detail mapping comprising:

means for compressing a texture map using blockwise two-level (one bit) quantization of brightness values or colors;

means for storing said compressed texture map on a storage medium;

means for mapping said stored texture map onto the surface of the computer generated object;

dedicated arithmetic unit means; and

memory units for storing at least one of texture, environment, reflectance and detail maps.

25. A method as set forth in claim 6, wherein combining said specularly reflected light intensity with a specular reflectance coefficient comprises multiplying said specularly reflected light intensity by the specular reflectance coefficient.

26. A method as set forth in claim 7, wherein combining the specularly reflected light intensity with the specular reflectance comprises multiplying the specularly reflected light intensity by the specular reflectance coefficient.

27. A method as set forth in claim 8, wherein a pointer into said detail map is assigned to each texture element of the texture map.

28. A method as set forth in claim 11, wherein at least one of an environment mapping, and a reflectance mapping ~~is~~ carried out in real time using dedicated arithmetic units.

29. A method as set forth in claim 21 wherein the texture is a reflectance map.

30. A method as set forth in claim 21 wherein the texture is a detail map.

31. The texturing unit of claim 16, wherein each block value represents the luminance of a texel.

32. The texturing unit of claim 16, wherein each block value represents an index into a look-up table.

33. The texturing unit of claim 16, wherein each block value represents the color of a texel.

34. A device for mapping real time video images onto a surface of a computer generated object, each video image comprising more than one scan-line, comprising;

a filter unit for generating prefiltered images of less detail; and

means for accessing pixels of a previous scan-line to perform said filtering.

35. A texturing unit for mapping a texture to a surface of a computer generated object, which texture comprises a plurality of blocks, each block comprising a plurality of texels and having two block values associated with the block, and each texel of each block corresponding to one of the two block values associated with the block, the texturing unit comprising:

a Random Access Memory (RAM) for storing the two block values associated with each block of the texture and a value for each texel, which value indicates the block value to which the texel corresponds;

a decompression unit coupled to the RAM, for accepting from the RAM values representing eight texels and the block values associated with each block of which the eight texels are part, and for determining eight decompressed texel values therefrom;

a trilinear interpolator coupled to the decompression unit, for accepting from the decompression unit the eight decompressed texel values and interpolating an interpolated value therefrom; and

an output port coupled to the trilinear interpolator, for transmitting the new value to a device coupled to the output port.

36. The texturing unit of claim 35, wherein the RAM is configured such that values for eight texels can be accessed substantially simultaneously, the eight texels comprising four texels from a first level and four texels from a second level, where the first level is one level higher than the second level.

37. The texturing unit of claim 36, wherein the four texels from the first level represent a two-by-two block of contiguous texels within the first level of the mipmap,

and the four texels from the second level represent a two-by-two block of contiguous texels within the second level of the mipmap.

38. The texturing unit of claim 36, wherein each decompressed texel value represents an index into a look-up table.

39. The texturing unit of claim 36, wherein each decompressed texel value represents the color of a texel.

40. The texturing unit of claim 35, wherein the RAM, the interpolator, and the output port are part of a single chip.

41. The texturing unit of claim 35, wherein the interpolator comprises at least one dedicated arithmetic unit.

42. The texturing unit of claim 41, wherein the RAM, the interpolator, and the output port are part of a single chip.

43. The texturing unit of claim 37, wherein the RAM, the trilinear interpolator, and the output port are part of a single chip.

44. The texturing unit of claim 43, wherein the trilinear interpolator comprises at least one dedicated arithmetic unit.

45. The texturing unit of claim 35, a ~~wherein the~~ texture comprises a plurality of blocks, each block comprising:
a plurality of texels and having two block values associated with the block, and each texel of each block corresponding to one of the two block values associated with the block, the information stored in the RAM comprising:
the two block values associated with each block of the texture; and
a value for each texel, which value indicates the block value to which the texel corresponds.

46. The texturing unit of claim 35, wherein each texel value represents the luminance of a texel.

47. The texturing unit of claim 35, wherein each texel value represents an index into a look-up table.

48. The texturing unit of claim 35, wherein each texel value represents the color of a texel.

49. The texturing unit of claim 35, wherein each decompressed texel value represents the luminance of a texel.

50. The texturing unit of claim 36, wherein the texture is a view of an environment of a scene.

51. The texturing unit of claim 36, wherein the texture is a reflectance map, and the texel values are specular reflectance coefficients.

52. The texturing unit of claim 36, wherein each texel is associated with a horizontal detail offset and a vertical detail offset, which horizontal detail offset and vertical detail offset are pointers into a detail map associated with the texture, which detail map is stored in the RAM.

53. The texturing unit of claim 52, wherein the detail map is a mipmap.

54. A texturing unit for mapping a texture to a surface of a computer generated object, which texture comprises a mipmap, which mipmap comprises a plurality of levels, each of which levels comprises ~~at least one~~ texel, the texturing unit comprising: a control unit for receiving an input signal and determining a set of N footprint texel locations and at least one footprint level of detail from the input signal, which input signal includes information about a location and a shape of a projection of a pixel on the texture;

a Random Access Memory (RAM) coupled to the control unit for, storing information representing the texture, receiving the set of N footprint texel locations and the footprint level of detail from the control unit, and determining N sets of texel values, where each set of texel values is associated with one footprint texel location, and where each set of texel values includes at least one texel value;

an interpolator coupled to the RAM, for accepting from the RAM the N sets of texel values and interpolating N interpolated values therefrom;

an averaging unit coupled to the interpolator for accepting from the interpolator the N interpolated values and determining an averaged value therefrom; and

an output port coupled to the averaging unit, for transmitting the averaged value to a device coupled to the output port.

55. The texturing unit of claim 54, further comprising:

a mipmap generation unit, coupled to the RAM, for accepting a changing video image, for generating a generated mipmap in real-time based on the changing video image, and for putting the generated mipmap into the RAM.

56. The texturing unit of claim 55, wherein the changing video image is an interlaced video image and the texturing unit further comprises:

a memory coupled to the mipmap generation unit for holding an interlaced half-frame of the interlaced video image.

57. The texturing unit of claim 55, wherein the mipmap generation unit calculates each level of the generated mipmap incrementally based on available information from the next level of higher detail.

add a17

Add 71